

Claims

1    1. A computer system having a data processing environment in  
2    which a program is divided into and executed as multiple threads,  
3    and in which said threads share and access data that is stored in  
4    a memory device, comprising:

5        means for indicating specific data that will be accessed  
6    only by a specific thread;

7        means for determining, when a thread attempts to access  
8    data, whether a specific thread indication is present relative to  
9    the data being accessed;

10        means for accessing said specific data without first  
11    performing a locking process to reject access attempts by other  
12    threads, when the specific thread indication is present; and

13        means for performing a locking process for the data being  
14    accessed before accessing the data when it is determined that no  
15    specific thread indication is present.

16        2. The computer system according to Claim 1, wherein said  
17    specific thread detects data, included in said data stored in  
18    said memory device, and said specific thread does not have a  
19    reference pointer to said data, and thereafter releases memory  
20    occupied by said data to provide storage space that is freely  
21    available.

1        3. In a data processing environment, a system in which multiple  
2    threads share and access objects, comprising:

3        flag data, provided for an object, for indicating an  
4    existence of a locality specifying that said object is to be  
5    accessed only by a specific thread;

6        means for having the specific thread access said object when  
7    said flag data for said object indicates said locality for said  
8    specific thread, without performing a locking process to reject  
9    access attempts by other threads or other objects before  
10   accessing said specific data; and

11        means for having the specific thread perform said locking  
12   process before accessing said object when said flag data does not  
13   indicate said locality for said specific thread.

14        4. The computer system according to Claim 3, wherein, when said  
15   object is created by a thread, said object sets said flag data  
16   indicating a locality exists for said thread, and wherein, before  
17   said object is changed so that it can be accessed by another  
18   thread or another object, said locality indicated by said flag  
19   data is canceled.

20        5. The computer system according to Claim 3, wherein said  
21   specific thread detects an object for which said flag data  
22   indicates the existence of a locality for said specific thread  
23   but said specific thread does not have a reference pointer to  
24   said data, and thereafter releases said object to provide in a  
25   memory device storage space that is freely available.

1       6. A memory management method for a data processing environment  
2       in which a program is divided into and executed as multiple  
3       threads, and in which said threads share and access objects that  
4       are stored in a memory device, comprising the steps of:

5               setting flag data indicating an existence of a locality for  
6       a specific object that is created by a specific thread and that  
7       is to be accessed only by said specific thread;

8

9               canceling said locality indicated by said flag data before  
10      said specific object is changed so that said specific object can  
11      be accessed by another thread;

12

13               without performing a locking process to reject access  
14      attempts by other threads or objects, accessing said specific  
15      object when said flag data for said specific object indicates the  
16      existence of a locality for said specific thread; and

17               locking said specific object before accessing said specific  
18      object when there is no said flag data indicating a locality for  
      said specific thread.

1       7. The memory management method according to Claim 6, wherein  
2       said step of canceling said locality indicated by said flag data  
3       for said specific object includes a step of:

4               performing said locking process, when said specific object  
5       has a locality for a specific thread, that was skipped at the  
6       time said specific object was accessed by said specific thread.

1       8. A memory management method for a data processing environment  
2       in which a program is divided into and is executed as multiple  
3       threads, and in which said threads share and access objects that  
4       are stored in a memory device, comprising the steps of:

5                setting flag data indicating an existence of a locality  
6       indicating that a specific object that is created by a specific  
7       thread is to be accessed only by said specific thread;

8                permitting said specific thread to detect an object for  
9       which flag data indicates the existence of a locality for said  
10      specific thread and said specific thread does not have a  
11      reference pointer to said object; and

12                releasing said detected object to provide additional storage  
13      space in the memory device that may be freely used.

14       9. Computer readable code stored on computer readable medium for  
15       permitting a locking step to be skipped in certain situations  
16       relative to data in a multi-thread environment, comprising:

17                first subprocesses for setting flag data indicating the  
18       existence of a locality for a specific object that is created by  
19       a specific thread and that is to be accessed only by said  
20       specific thread;

21                second subprocesses for canceling said locality indicated by  
22       said flag data before said specific object is changed so that  
23       said specific object can be accessed by another thread;

11       third subprocesses accessing said specific object when said  
12 flag data for said specific object indicates the existence of a  
13 locality for said specific thread without performing a locking  
14 process to reject access attempts by other threads; and

15       forth subprocesses for performing said locking process  
16 before accessing said specific object when said flag data  
17 indicates the absence of a locality for said specific thread.

1       10. Computer readable code stored on computer readable medium  
2 for performing memory management for a program that executes in  
3 multiple threads, comprising:

4       first subprocesses for setting flag data indicating  
5 existence of a locality indicating that a specific object that is  
6 created by a specific thread is to be accessed only by said  
7 specific thread;

8       second subprocesses for permitting said specific thread to  
9 detect an object for which flag data indicates the existence of a  
10 locality for said specific thread and said specific thread does  
11 not have a reference pointer to said object; and

12       third subprocesses for unlocking said detected object so  
13 that storage space may be freely used.